The Flask Security Architecture: System Support for Diverse Security Policies

Ray Spencer (Secure Computing Corporation)
Stephen Smalley, Peter Loscocco (National Security Agency)
Mike Hibler, David Andersen, Jay Lepreau (University of Utah)

Univ. of Utah Technical Report UUCS-98-014
August 1998

Abstract

Operating systems must be flexible in their support for security policies, i.e., the operating system must provide sufficient mechanisms for supporting the wide variety of real-world security policies. Systems claiming to provide this support have failed to do so in two ways: they either fail to provide sufficient control over the propagation of access rights, or they fail to provide enforcement mechanisms to support fine-grained control and dynamic security policies. In this paper we present an operating systems security architecture that solves both of these problems. The first problem is solved by ensuring that the security policy (through a consistent replica) is consulted for every security decision. The second problem is solved through mechanisms that are directly integrated into the service-providing components of the system. The architecture is described through its prototype implementation in the Flask microkernel-based OS, and the policy flexibility of the prototype is evaluated. We present initial evidence that the architecture’s performance impact is modest. Moreover, our architecture is applicable to many other types of operating systems and environments.

1 Introduction

A phenomenal growth in connectivity through the Internet has made computer security a paramount concern, but no single definition of security suffices. Different computing environments, and the applications that run in them, have different security requirements. Because any notion of security is captured in the expression of a security policy, there is a need for many different policies, and even many types of policies [1, 45, 49]. To be generally acceptable, any computer security solution must be flexible enough to support this wide range of security policies. This flexibility must be supported by the security mechanisms of the operating system.

Supporting flexibility in the operating system is a hard problem that goes beyond just the supporting of multiple policies. The system must be capable of supporting fine-grained access controls on low-level objects used to perform higher-level functions controlled by the security policy. Additionally, the system must ensure that the propagation of access rights is in accordance with the security policy. Lastly, policies are not, in general, static. To cope with policy changes or dynamic policies, there must be a facility for the revocation of previously granted access rights. Other systems that claim to support policy flexibility fail to adequately address at least one of these three areas.

This paper describes an operating system security architecture that demonstrates the feasibility of policy flexibility. This is done by presenting its prototype implementation, the Flask microkernel-based OS, that successfully overcomes the hard problems of policy flexibility where other systems have failed. The cleaner separation of mechanism and policy specified in the security architecture facilitates policy flexibility. Flask includes a security policy server to make access control decisions and a policy-flexible enforcement framework in the microkernel and other object managers in the system. Although the prototype system is microkernel-based, the security mechanisms do not depend on a microkernel architecture and will easily generalize beyond it.

The resulting system provides policy flexibility. It supports a wide variety of policy types. It controls the propagation of access rights by ensuring that the security policy, through a consistent replica, is consulted for every access decision. Enforcement mechanisms directly integrated into the service-providing components of the system enable fine-grained access controls and dynamic policy support that allows the revocation of previously granted access rights. Initial performance results indicate that the impact of policy flexible security on the system can be kept to a minimum.

The remainder of the paper begins by elaborating on the meaning of policy flexibility. After a discussion of why two popular mechanisms employed in systems that purport to be policy flexible are really limiting to policy flexibility, some